Problem 22: Free Up Disk Space

Points: 60

Author: Doug Kelley, Palmdale, California, United States

Problem Background

The amount of disk space on your computer is getting low. We need an algorithm to archive the oldest files (because they probably aren't being used any more) and the biggest files (because they take up the most space).

Remember that a kilobyte (KB) is 1,000 bytes (B). A megabyte (MB) is 1,000 KB. A gigabyte (GB) is 1,000 MB.

Problem Description

Your program will be given a list of files on your computer and some information about them. Each file should be assigned a score based on its age and size. Using today's date of April 27th, 2019, determine the age of the file in days. A file created in the morning (from 12:00 AM through 11:59 AM) should be counted as ½ day older than one created the same day in the afternoon (12:00 PM through 11:59 PM); thus, a file last modified yesterday at 1:00 PM is 0.5 days old; one modified yesterday morning is 1.0 days old. Multiply the file's age in days by its size in MB to determine the file's score. Remember to account for leap years in your calculations.

For example, a 1500 KB file was last modified on April 27, 2018, at 10:00 PM. The file is 364.5 days old (365 days, minus 0.5 days for an afternoon time). Multiplying this value

by the file's size in MB - 1500 KB = 1.5 MB - results in a score of 546.75.

Your program must list the highest-scoring files and their scores until the total size of the listed files accounts for at least 25% of your hard drive's capacity.

Sample Input

The first line of your program's input, **received from the standard input channel**, will contain a positive integer representing the number of test cases. Each test case will include the following lines of input:

- A line containing a positive integer, **F**, indicating the number of files on your computer, a space, and a positive decimal, **C**, indicating the size of your hard drive in GB.
- **F** lines listing the following information. Each data point is separated by spaces.
    - The date the file was last modified, in DD/MM/YYYY format. All dates will be no later than April 26, 2019.
    - The time the file was last modified, in HH:MM format

    - "AM" or "PM", indicating if the timestamp was in the morning or afternoon
    - A positive integer representing the size of the file in KB
    - The name of the file. File names may include uppercase and lowercase letters, numbers, and periods (.).

```
1
10 1.0
25/04/2019 10:30 AM 125000 file1.txt
02/03/2019 02:15 PM 62500 file2.exe
01/01/2019 05:34 PM 62500 file3.mov
31/12/2018 11:36 AM 31250 file4.gif
14/02/2019 10:42 PM 31250 file5.doc
23/08/2018 12:00 PM 31250 file6.sh
29/02/2016 09:20 AM 31250 file7.mp3
05/12/2018 01:30 PM 15625 file8
26/04/2019 01:30 PM 15625 file9.png
01/01/2000 04:15 PM 1000 file10.jpg
```

Sample Output

For each test case, your program must output the information listed below for the highest-scoring files. Continue listing files until the total size of the listed files is equal to or greater than 25% of **C**.

- The name of the file
- A space
- The score calculated for that file, rounded to three decimal places. Include any trailing zeroes.

file7.mp3 36031.250
file6.sh 7703.125
file3.mov 7218.750
file10.jpg 7055.500
file4.gif 3656.250
file2.exe 3468.750
file5.doc 2234.375